CHAPTER 7: CHECK POINT QOS

Quality of Service (QoS) is a set of intelligent network protocols and services used to efficiently manage the movement of information through local and wide-area networks. QoS allows Security Administrators to prioritize traffic flows and provide better service to certain flows. Check Point QoS is included in every NGX product installation.

Objectives

Given a variety of Check Point QoS configurations, determine how to allocate bandwidth:

1. Determine if Check Point QoS is an appropriate solution, given a variety of business scenarios.
2. Configure Check Point QoS to meet the requirements, given a variety of business requirements.
3. Determine how bandwidth will be allocated, given a variety of Check Point QoS configurations.
4. Identify situations where Low Latency Queueing and Differentiated Services are an appropriate part of a QoS solution.

Key Terms

- Check Point QoS
- IQ Engine
- Weighted Flow Random Early Drop (WFRED)
- Retransmission Detection Early Drop (RDED)
- Differentiated Services (DiffServ)
- Weighted Fair Queuing (WFQ)
- Low Latency Queuing

CHECK POINT QOS OVERVIEW

Check Point QoS is a bandwidth-management solution used to alleviate bandwidth congestion at network-access points. Check Point QoS controls both inbound and outbound traffic flows. Check Point QoS uses four technologies to control traffic:

- Stateful Inspection
- Intelligent Queuing Engine (IQ Engine)
- Weighted Flow Random Early Drop (WFRED)
- Retransmission Detection Early Drop (RDED)

STATEFUL INSPECTION

Check Point QoS uses Check Point's patented Stateful Inspection technology to derive complete state and context information for all network traffic.

INTELLIGENT QUEUING ENGINE (IQ ENGINE)

Check Point QoS's **IQ Engine** uses state-derived information to classify traffic and place it in the proper transmission queue. Check Point QoS uses a packet scheduler to move packets through a dynamically changing scheduling tree at different rates, in accordance with the QoS Policy.

WEIGHTED FLOW RANDOM EARLY DROP (WFRED)

Weighted Flow Random Early Drop (WFRED) is a mechanism for managing packet buffers, by selectively dropping packets during periods of network congestion. WFRED is transparent to users and requires no configuration.

RETRANSMISSION DETECTION EARLY DROP (RDED)

Retransmission Detection Early Drop (RDED) is a mechanism for reducing the number of retransmits and retransmit storms during periods of network congestion. RDED prevents inefficiencies by detecting retransmits in TCP streams and preventing the transmission of redundant packets when multiple copies of a packet are concurrently queued on the same flow.

Check Point QoS Architecture

The architecture and flow of Check Point QoS is similar to VPN-1 NGX. Check Point QoS has three components:

SmartConsole — Configures and monitor Check Point QoS

SmartCenter Server — Stores and distribute the Check Point QoS Policy

QoS Module — Enforces Check Point QoS Policy

Like other NGX products, Check Point QoS components can be installed in stand-alone or distributed deployments.

SMARTCONSOLE AND CHECK POINT QOS

The SmartConsole suite of tools allows a Security Administrator to configure, verify, install, and monitor the Check Point QoS Policy. SmartDashboard is used to configure the Policy. SmartView Tracker, SmartView Monitor, and Eventia Reporter provide useful information that can be used to monitor and tune the Policy.

The QoS information in SmartView Monitor is provided by interface. When reviewing Traffic > Top QoS Rules, and interface must be selected. If an interface is not selected, SmartView Monitor will display an error message.

Check Point's integrated architecture allows Administrators to reuse objects created for Security Policies when defining QoS Policies in SmartDashboard. A single SmartCenter Server can control and monitor multiple QoS modules.

SMARTCENTER SERVER AND CHECK POINT QOS

After the QoS Policy is defined in SmartDashboard, it is downloaded to the SmartCenter Server. The SmartCenter Server verifies the Policy and distributes it to QoS Modules. Policy distribution is handled by the Check Point Daemon (CPD), which runs on both the SmartCenter Server and QoS Module. The SmartCenter Server also manages the Check Point Log Repository, and acts as a log server for SmartView Tracker.

QOS MODULE

The QoS Module enforces the QoS Policy and provides log information to the SmartCenter Server. The QoS Module's functions are divided between the QoS kernel driver and QoS daemon. The QoS kernel driver examines, queues, schedules, and releases packets. The QoS daemon is a user-mode process that performs tasks that are difficult for the kernel. The QoS daemon performs the following tasks for the kernel:

- DNS name resolution
- Resolving authenticated data for an IP (UserAuthority integration)
- Updating the kernel regarding cluster status (in a Load Sharing configuration)

The Check Point QoS module must be installed with VPN-1 NGX. Both products share similar architecture, and Check Point QoS relies on NGX technology to perform its bandwidth-management functions.

Check Point QoS Deployment Considerations

A distributed deployment is strongly recommended for Check Point QoS. Distributed deployments scale gracefully, and dedicated servers reduce the likelihood that competition for server resources will create a bottleneck. CPU performance is the main factor affecting the performance of Check Point QoS modules. A server-class system is strongly suggested. Memory requirements are based on the number of connections the Check Point QoS Module is expected to manage. The table below provides guidelines for memory. On average, each connection requires 1,300 bytes of memory.

Number of Connections	Suggested Memory
10,000	39 MB
25,000	57 MB
50,000	91 MB
100,000	156 MB

CHECK POINT QOS POLICY

Defining a Check Point QoS Policy is very similar to defining a Security Policy. Rules and objects define the actions a QoS Module should take for different types of traffic. In the QoS Policy, connections are classified according to four criteria:

Source — Nodes, networks, user groups, domains

Destination — Nodes, networks, user groups, domains

Resources — Only resources of type URI for QoS

Service — Limited to:

- TCP
- Compound TCP
- UDP
- ICMP
- Citrix TCP Services
- IP Services

Time — Defined for specific times and/or specific days

These fields can be populated with the same objects used to define Security and Desktop Policies. The Action configuration options for the QoS Policy differ significantly from the actions available in a Security or Desktop Policy. The Action field in a QoS rule is used to:

- Define Weights.
- Set Limits.
- Set Guarantees.

Other QoS parameters are configured on the QoS Modules' Topology > Interface Properties screens and on the Global Properties > QoS screen.

Check Point QoS Rule Base

The Check Point QoS Rule Base can be configured in either Express or Traditional mode. Express mode allows an Administrator to define basic Policies quickly and easily. Traditional mode incorporates the more advanced features of Check Point QoS.

EXPRESS MODE FEATURES

The following Check Point QoS features are available in Express mode:

- Weights
- Per-rule guarantees and limits
- Logging and accounting
- Hardware-accelerator support
- High Availability and Load Sharing

TRADITIONAL MODE FEATURES

In addition to all the features available in Express mode, Traditional mode also has the following features available:

- Authenticated QoS (UserAuthority integration)
- Per-connection guarantees and limits
- Low Latency Queuing
- Differentiated Services support
- Sub-rules
- Matching by URI resources
- Matching by DNS string
- TCP Retransmission Detection (RDED) mechanism
- Matching Citrix ICA applications

QoS Action Properties

The fields in QoS rules can be thought of as defining:

- Who or what:
  - — Source
  - — Destination
  - — Service
- When:
  - — Time
- How:
  - — Action

The Action fields in the QoS rules determine how bandwidth will be allocated. Most actions allow for dynamic allocation of bandwidth, based on current connections. Low Latency Queuing is an exception, and is discussed later in this chapter.

SIMPLE AND ADVANCED ACTION TYPES

When an Administrator edits the Action field of a QoS rule, either Simple or Advanced Action Types may be selected. The configuration options for Simple Action type allow an Administrator to:

- Apply the rule only to encrypted traffic.
- Set a weight for the rule.
- Set a limit for the rule.
- Set a guarantee for the rule.

The Advanced action type selection adds configuration options for setting per-connection limits and guarantees.

The Advanced action type option is only available for Traditional mode, not Express mode. If per-connection limits and guarantees are needed, convert the Policy to Traditional mode, using the Policy > Convert To option from the main menu.

WEIGHTS

Weights are used to set the relative priority for different traffic flows. Weight is the relative portion of available bandwidth allocated to a rule or connection. Bandwidth is allocated dynamically, and only open connections are included in the calculation. The formula for determining a rule's portion of the available bandwidth is:

this rule's portion = this rule's weight / total weight of all rules with open connections

Weights can only be set per rule, not per connection.

GUARANTEES

A guarantee allocates a minimum amount of bandwidth to the connections matched with a rule. Guarantees can be defined per rule and per connection. When guarantees are specified per connection, the Administrator should also determine how many connections will be permitted. Weights ensure proportional shares, but only guarantees allow an Administrator to specify an absolute bandwidth value. Guaranteed bandwidth is allocated before bandwidth is distributed according to weights.

LIMITS

Limits define the maximum bandwidth that can be assigned to connections matching a rule. A limit defines a point beyond which connections under a rule are not allocated bandwidth. A limit can be defined for the entire rule, thus setting an absolute maximum quantity of bandwidth the rule may consume. Limits can also be set for each connection matched with a rule.

DEFAULT RULE

A default rule is automatically added to each QoS Policy Rule Base. The weight of the default rule is defined on the QoS screen in Global Properties. The default rule applies to all connections not matched by other rules or sub-rules. The default rule can be modified, but it cannot be deleted.

Bandwidth Allocation and Rules

The following sections provide examples of bandwidth allocation and rules. Each example includes QoS Rule Base information, open connections, available bandwidth, and an explanation of how to determine how much bandwidth a particular connection will be allocated.

Check Point QoS ensures optimal bandwidth use, by allocating bandwidth dynamically. Each example provided is a “snapshot”. In a live network, connections open and close constantly. By the time an Administrator figures out per-connection allocation, the allocations will have changed.

SIMPLE QOS RULE BASE WITH WEIGHTS ONLY

Adam is the Administrator at ABC Company. ABC Company has only two networks, Execu_Net and Other_Net. Adam has been told by upper management that traffic from Execu_Net needs to be given priority over traffic from Other_Net.

Adam configures a simple QoS Policy with the following rules:

Name	Executive Net Rule	Default
Source	Execu_Net	Any
Destination	Any	Any
Service	Any	Any
Action	Weight 30	Weight 10
Track	Log	Log
Install On	Any	Any
Time	Any	Any
Comment	Give Execu_Net Priority	Blank

The total bandwidth available to the QoS Module is 190 KBps.

The current open connections on the QoS Module are:

- 8 HTTP connections from Execu_Net.
- 3 SMTP connections from Other_Net.
- 2 RealAudio connections from Execu_Net.

Bandwidth is distributed among the open connections according to weight, using the formula:

this rule's portion = this rule's weight / total weight of all rules with open connections

In this example, connections match both rules, so the total weight of all rules with open connections is 40 (Executive Net Rule = 30, Default Rule = 10).

The bandwidth allocated to the Executive Net Rule is:

(30/40) * 190 = 142.5 KBps

The 142.5 KBps allocated to the Executive Net Rule is shared among the 8 HTTP sessions and the 2 RealAudio sessions.

The bandwidth allocated to the Default Rule is:

(10/40) * 190 = 47.5 KBps

The 47.5 KBps is shared among the 3 SMTP sessions from Other_Net.

Q.) How would bandwidth be allocated, if all connections originated from Execu_Net?

A.) All 190 KBps would be shared among the connections from Execu_Net. If no open connections match a QoS rule, that rule is not allocated any bandwidth.

Q.) How would bandwidth be allocated, if no open connections matched the Executive Net Rule?

A.) All bandwidth would be allocated to the Default Rule.

QOS RULES BY SOURCE AND SERVICE WITH WEIGHTS ONLY

Bonnie is the Administrator at DEF Corporation. DEF Corporation uses IMAP as part of their e-mail solution. Bonnie has been instructed to give priority first to IMAP traffic, and second to the marketing network. All other traffic can be allocated bandwidth by the Default Rule.

Bonnie configures a QoS Policy with the following rules:

Name	IMAP Rule	Marketing Rule	Default Rule
Source	Any	Marketing_Net	Any
Destination	Any	Any	Any
Service	IMAP	Any	Any
Action	Weight 50	Weight 20	Weight 10
Track	None	None	None
Install On	Any	Any	Any
Time	Any	Any	Any
Comment	Prefer IMAP	Blank	Blank

The total bandwidth available to the QoS Module is 260 KBps.

The current open connections on the QoS Module are:

- 23 IMAP connections from Engineering_Net.
- 32 IMAP connections from Marketing_Net.
- 57 HTTP connections from Marketing_Net.

In this example, only the IMAP and Marketing Rules have connections that match, so the Default Rule is not used in calculating the total weight of all rules with open connections. The total weight of all rules with open connections is 70 (IMAP Rule = 50, Marketing Rule = 20).

Traffic is processed on the first QoS rule it matches. In this example, the 32 IMAP connections from Marketing_Net and the 23 IMAP connections from Engineering_Net all match the IMAP Rule. The 57 HTTP connections from Marketing_Net match the Marketing Rule.

The bandwidth allocated to the IMAP Rule is:

(50/70) * 260 KBps = 185.7 KBps

The 185.7 KBps allocated to the IMAP Rule is shared among the 55 connections matching that rule.

The bandwidth allocated to the Marketing rule is:

(20/70) * 260 KBps = 74.3 KBps

The 74.3 KBps allocated to the Marketing Rule is shared among the 57 connections matching the rule.

Q.) How would the bandwidth allocation change if all other connections stayed open, but an additional connection was opened from the Engineering_Net using the Telnet service?

A.) The Telnet session from the Engineering_Net would match the Default Rule. Check Point QoS would dynamically reallocate the bandwidth to provide the Default Rule its portion of bandwidth. The total weight of all rules with open connections becomes 80. Bandwidth is allocated by weight:

(50/80) * 260 KBps = 162.5 KBps for the IMAP Rule

(20/80) * 260 KBps = 65 KBps for the Marketing Rule

(10/80) * 260 KBps = 32.5 KBps for the Default Rule

Q.) How is the bandwidth divided, by connection, in the question above?

A.) The connections matching each rule share the allocated bandwidth, so:

55 connections share the 162.5 KBps allocated to the IMAP Rule.

57 connections share the 65 KBps allocated to the Marketing Rule.

1 connection gets the entire 32.5 KBps allocated to the Default Rule.

QOS RULES BY SOURCE AND SERVICE WITH WEIGHTS AND GUARANTEES

Carl is the Administrator at GHI Corporation. Upper management at GHI Corporation instructed Carl to ensure that 20 percent of the available bandwidth is guaranteed for RealAudio connections. IMAP connections and connections from the Executive network should also receive priority. IMAP connections are a higher priority than connections from the Executive network.

Carl configures a QoS Policy with the following rules:

Name	RealAudio	IMAP	Execu_Net	Default
Source	Any	Any	Execu_Net	Any
Destination	Any	Any	Any	Any
Service	RealAudio	IMAP	Any	Any
Action	Weight 10 Guarantee 100 MBps	Weight 30	Weight 20	Weight 10
Track	None	None	None	None
Install On	Any	Any	Any	Any
Time	Any	Any	Any	Any
Comment	Blank	Blank	Blank	Blank

The total bandwidth available to the QoS Module is 500 MBps.

The current open connections on the QoS Module include the following:

- 7 RealAudio connections from Execu_Net
- 3 RealAudio connections from Marketing_Net
- 298 IMAP connections from Operations_Net
- 11 IMAP connections from Execu_Net
- 22 Telnet connections from Engineering_Net
- 8 IMAP connections from Engineering_Net
- 9 HTTP connections from Execu_Net
- 40 HTTP connections from Operations_Net
- 3 HTTP connections from Engineering_Net

Bandwidth is allocated to rules with guarantees before it is distributed by weights. Allocation by guarantee only happens, if there is an open connection matching the rule. In this example, the RealAudio Rule has a guarantee, and there are 10 open connections matching that rule. First the QoS Module allocates 100 MBps to the RealAudio Rule. The remaining 400 MBps is divided among all rules with open connections, including the RealAudio Rule, by weight.

All four rules have matching connections, so the total weight of all rules with open connections is 70. Bandwidth distribution by weight occurs next:

[(10/70) * 400 MBps] + 100 MBps = 157.1 MBps for the RealAudio Rule

(30/70) * 400 MBps = 171.4 MBps for the IMAP Rule

(20/70) * 400 MBps = 114.3 MBps for the Execu_Net Rule

(10/70) * 400 MBps = 57.1 MBps for the Default Rule

Q.) How many connections are sharing the Execu_Net Rule's 114.3 MBps?

A.) Nine HTTP connections match the Execu_Net Rule. The RealAudio and IMAP connections from the Execu_Net network matched on earlier rules.

Q.) If all RealAudio connections are terminated and all other connections remain open, how is the bandwidth redistributed?

A.) The guarantee is not implemented, and the RealAudio Rule's weight is not included in the total weight of all rules with open connections, thus:

(30/60) * 500 MBps = 250 MBps for the IMAP Rule

(20/60) * 500 MBps = 166.7 MBps for the Execu_Net Rule

(10/60) * 500 MBps = 83.3 MBps for the Default Rule

QOS RULES BY SERVICE WITH WEIGHTS AND PER-CONNECTION GUARANTEES

Debbie is the Administrator at JKL Corporation. JKL Corporation uses H.323 for video conferencing. Analysis of the H.323 traffic on her network and a review of the video-conferencing software have led Debbie to the conclusion that each H.323 connection requires 384-512 KBps. No more than four H.323 connections will be open concurrently. So Debbie chooses not to allow additional connections beyond the four permitted in the per connections setting in the rule's Action properties. All other connections are of equal priority.

Debbie configures a QoS Policy with the following rules:

Name	Video_Conferencing	Default
Source	Any	Any
Destination	Any	Any
Service	H.323	Any
Action	Weight 1 PC 512 KBps	Weight 10
Track	None	None
Install On	Any	Any
Time	Any	Any
Comment	Blank	Blank

The total bandwidth available to the QoS Module is 8 MBps (8,192 KBps).

If the four H.323 connections are open and there are traffic flows matching the Default Rule, bandwidth will be allocated first to the per-connection guarantees and then by weight. Each H.323 connection will be allocated 512 KBps, and the remaining bandwidth will be distributed by weight, across all rules, including the H.323 Rule.

8,192 - (4 * 512) = 6,144 KBps to distribute by weight

[(1/11) * 6,144] + (4 * 512) = 2,662.4 KBps allocated to the H.323 Rule

(9/11) * 6,144 = 5,529.6 KBps shared by all flows matching the Default Rule

Q.) If all four H.323 connections remained open, and the traffic flows matching the Default Rule all closed, how would bandwidth be allocated?

A.) The remaining 6,144 KBps would be shared among the four connections matching the H.323 Rule.

Q.) Since Debbie did not allow additional connections in the Action properties, what happens if someone attempts to establish a fifth H.323 connection?

A.) No bandwidth will be allocated to the fifth H.323 connection.

If Debbie checks Accept Additional Connections in the H.323 Rule's Action properties, bandwidth will be allocated to additional connections by weight. If four H.323 connections have guaranteed bandwidth of 512 KBps, and two additional connections are opened and connections match the Default Rule, bandwidth will be allocated:

1. The four guaranteed H.323 connections will be given their 512 KBps.
2. The Default Rule will receive its 10/11 of the remaining bandwidth (5,529.6 KBps).
3. The remaining 1/11 of the bandwidth (614.4 KBps) will be shared among the six H.323 connections:

   614.4 KBps for the first 4 H.323 connections and

   102.4 KBps for the next 2 H.323 connections

Since Debbie's research indicates 384-512 KBps are required to sustain minimum quality for the H.323 sessions, she should consider either increasing the weight of the H.323 Rule or not allowing additional connections.

The sum of guarantees in rules in the upper level should not exceed 90 percent of the capacity of the link.

QOS BY SOURCE AND SERVICE WITH LIMITS AND WEIGHTS

Edward, the Administrator for MNO Corporation, must ensure that the total bandwidth consumed by FTP traffic never exceeds 250 KBps. Also, traffic from Execu_Net should receive higher priority than other traffic.

Edward configures a QoS Policy with the following rules:

Name	FTP_Limit	Execu_Net	Default
Source	Any	Execu_Net	Any
Destination	Any	Any	Any
Service	FTP	Any	Any
Action	Weight 10 Limit 250 KBps	Weight 20	Weight 10
Track	None	None	None
Install On	Any	Any	Any
Time	Any	Any	Any
Comment	Blank	Blank	Blank

The total bandwidth available to the QoS Module is 2 MBps (2,048 KBps).

If connections match all three rules in the QoS Policy, bandwidth will be allocated, as follows:

250 KBps for the FTP_Limit Rule, because its allocation cannot exceed the defined limit

(20/30) * 1,798 = 1,198.7 KBps for the Execu_Net Rule

(10/30) * 1,798 = 599.3 KBps for the Default Rule

If the bandwidth allocated by weight is less than the limit, the rule will receive the lower allocation. In the example above, if Edward's QoS Module had only 512 KBps total available bandwidth, and all three rules had open connections matching them, the FTP_Limit Rule would have been allocated 128 KBps.

QOS BY SOURCE WITH WEIGHTS AND PER-CONNECTION LIMITS

Edward analyzes the traffic at MNO Corporation, and determines that limiting each FTP connection to no more than 20 KBps will improve his compliance directives from management. He changes the FTP_Limit Rule to read as follows:

Name	FTP_Limit
Source	Any
Destination	Any
Service	FTP
Action	Weight 10 Limit 250 KBps LC 20 KBps
Track	None
Install On	Any
Time	Any
Comment	Blank

Now any single FTP connection will be able to consume no more than 20 KBps, and the sum total of all FTP connections will be able to consume no more than 250 KBps.

When per-connection limits are imposed, the number of connections matching the rule with per-connection limits has an impact on the total bandwidth remaining for allocation by weight. For example, if only two FTP connections were open, then only 40 KBps would be allocated to the FTP_Limit Rule. The remaining 2,008 KBps would be distributed among connections matching the Execu_Net and Default Rules as follows.

GUARANTEE AND LIMIT INTERACTION

If a rule limit and guarantee per rule are defined in a rule, the limit should not be smaller than the guarantee. A combination of guarantees and limits can be used instead of very low weights, to ensure connections get the bandwidth they need, but do not consume a large amount of the bandwidth allocated in the weight distribution.

Recall the example with Debbie and JKL Corporation. Debbie's research indicated that each H.323 connection needed from 384-512 KBps to sustain optimal quality. Debbie could use a per-connection guarantee to make sure each H.323 connection received at least 384 KBps, and a per-connection limit to make sure none received more than 512 KBps.

Debbie configures a QoS Policy with the following rules:

Name	Video_Conferencing	Default
Source	Any	Any
Destination	Any	Any
Service	H.323	Any
Action	Weight 10 PC 384 KBps LC 512 KBps	Weight 10
Track	None	None
Install On	Any	Any
Time	Any	Any
Comment	Blank	Blank

The total bandwidth available to the QoS Module is 8 MBps (8,192 KBps).

Debbie only provides the per-connection guarantee for four connections, but she allows additional connections in the Action properties. Notice that she also changed the weight for the Video_Conferencing Rule from 1 to 10, so it is now equal to the weight for the Default Rule. If there are connections matching both the Video_Conferencing Rule and the Default Rule, and there are six H.323 sessions open, bandwidth will be allocated as follows:

1. 384 KBps to each of the first four H.323 sessions (total of 1,536 KBps)
2. All six H.323 connections will be taken up to 512 KBps, provided it does not exceed the bandwidth allocation by weight. In this case, the by-weight allocation would have been 3,328 KBps, but only 1,536 KBps is required to take each connection to its 512 KBps limit.
3. The remaining 5,120 KBps is allocated to open connections matching the Default Rule.

BANDWIDTH ALLOCATION AND SUB-RULES

Sub-rules are rules within a rule. Sub-rules allow an Administrator even more granular control over bandwidth allocation, but permit weights, guarantees, and limits within top-level rules. Sub-rules are only available in Traditional mode in the Rule Base.

Add Sub-Rule — QoS only, to add sub-rules to QoS rules

Add QoS Class — Adds a QoS Class of Service to a QoS rule

When a sub-rule is created for a top-level rule, a default rule for the sub-rule set is also generated. The default rule for the sub-rule set determines the fate of traffic that does not match any of the defined sub-rules in the set.

George is the Administrator at PQR Corporation. PQR Corporation provides a wide variety of services to its customers, via Web applications over HTTP. George needs to ensure that HTTP traffic receives higher priority than all other traffic, but he also must also make sure that inbound HTTP traffic receives higher priority than outbound HTTP traffic.

George configures a QoS Policy with the following rules:

Name	Web_Traffic	Default
Source	Any	Any
Destination	Any	Any
Service	HTTP	Any
Action	Weight 20	Weight 10
Track	None	None
Install On	Any	Any
Time	Any	Any
Comment	Blank	Blank

The total bandwidth available to the QoS Module is 8 MBps (8,192 KBps).

George creates a network object “PQR_Net”, to represent the supernet that encompasses all of PQR’s internal networks. George then configures two sub-rules for the Web_Traffic Rule. (The Default Rule for the Web_Traffic sub-rule set was added automatically by Check Point QoS, when the first sub-rule was created.)

Name	Inbound_Web	Outbound_Web	Default
Source	Any	PQR_Net	Any
Destination	PQR_Net	Any	Any
Service	Any	Any	Any
Action	Weight 50	Weight 20	Weight 10
Track	None	None	None
Install On	Any	Any	Any
Time	Any	Any	Any
Comment	Blank	Blank	Blank

There is no need to define the service for these sub-rules, as only HTTP traffic with be passed down from the top-level rule. The Source, Destination, and Service fields of a sub-rule must always be a subset of the parent rule, or the sub-rule will not be effective.

Connections match both the top-level rules (Web_Traffic and Default), so bandwidth is allocated between the two rules according to weight:

(20/30) * 8,192 = 5,461.3 KBps for the Web_Traffic Rule

(10/30) * 8,192 = 2,730.7 KBps for the Default Rule

Connections match all three sub-rules of the Web_Traffic Rule. The bandwidth allocated to the Web_Traffic Rule is further subdivided, according to the weights of the sub-rules:

(50/80) * 5,461.3 = 3,413.3 KBps for the Inbound_Web sub-rule

(20/80) * 5,461.3 = 1,365.3 KBps for the Outbound_Web sub-rule

(10/80) * 5,461.3 = 682.7 KBps for the Default sub-rule

Guarantees and limits within sub-rule sets are handled the same way as with top-level rules. The bandwidth is limited to that allocated to the sub-rule set's top-level rule.

ADDITIONAL QOS RULE CONSIDERATIONS

Some additional considerations for QoS rules include:

- If a guarantee is defined in a sub-rule, a guarantee must be defined for the rule above it.
- The guarantee of a sub-rule cannot be greater than the guarantee defined for the rule above it.
- A rule guarantee must not be smaller than the sum of guarantees defined in its sub-rules.
- If a rule's weight is low, some connections may receive very little bandwidth.
- The sum of guarantees in rules in the upper level should not exceed 90 percent of the capacity of the link.
- If per-connection guarantees are defined both for a rule and its sub-rules, the sum of the per-connection guarantees for the sub-rules should not be greater than the per-connection guarantee of the top-level rule.

- If both a rule and per-connection limit are defined for a rule, the per-connection limit must not be greater than the rule limit.
- If a limit is defined in a rule with sub-rules and limits are defined in all sub-rules, the rule limit should not be greater than the sum of limits defined in the sub-rules.
- If a rule limit and a guarantee per rule are defined in a rule, the limit should not be smaller than the guarantee.

If both a limit and a guarantee are defined in a rule and the limit is equal to the guarantee, connections may receive no bandwidth. This situation can occur when the rule has sub-rules with total rule guarantees that add up to the total rule guarantee for the entire rule, and the rule also has sub-rules with no guarantee.

DIFFERENTIATED SERVICES

Differentiated Services (DiffServ) is an architecture for providing different types or levels of service for network traffic. Inside the enterprise network, packets are marked in the IP header TOS byte as belonging to a certain class of service, or QoS class. These packets are then granted priority on the public network. DiffServ packets have meaning on the public network, not inside the enterprise network. Effective implementation of DiffServ requires that packet markings be recognized on all public network segments.

DiffServ Marks for IPSec Packets

When DiffServ marks are used for IPSec packets, the DiffServ mark can be copied from one location to another in one of two ways:

```
:ipsec.copy_TOS_to_inner
```

The DiffServ mark is copied from the IPSec header to the IP header of the original packet, after decapsulation/decryption:

```
:ipsec.copy_TOS_to_outer
```

The DiffServ mark is copied from the original packet's IP header to the IPSec header of the encrypted packet, after encapsulation.

This property should be set, per QoS Module, in `$FWDIR/conf/objects_5_0.c`.

The default setting is:

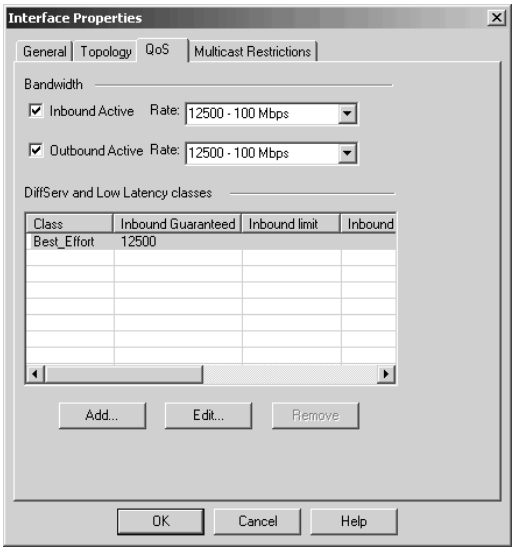
```
:ipsec.copy_TOS_to_inner (false)

:ipsec.copy_TOS_to_outer (true)
```

Interaction between DiffServ Rules and Other Rules

A DiffServ rule specifies not only a QoS class, but also a weight, in the same way that other QoS Policy rules do. These weights are enforced only on the interfaces on which the rules of this class are installed.

For example, suppose a DiffServ rule specifies a weight of 50 for FTP connections. That rule is installed only on the interfaces for which the QoS class is defined.



Interface QoS Properties Tab

On other interfaces, the rule is not installed and FTP connections routed through those other interfaces do not receive the weight specified in the rule. To specify a weight for all FTP connections, add a rule under “Best Effort.”

DiffServ rules can be installed only on interfaces for which the relevant QoS class has been defined in the QoS tab of the Interface Properties screen. “Best Effort” rules (that is, non-DiffServ rules) can be installed on all interfaces of Gateways with QoS Modules installed. Only rules installed on the same interface interact with each other.

LOW LATENCY QUEUING

For most traffic, **Weighted Fair Queuing** (WFQ) is adequate. WFQ helps avoid drops caused by congestion. Avoiding drops can means holding long queues. Long queues may lead to non-negligible delays.

Long queues are inappropriate for voice and video traffic. For most delay-sensitive applications, packets need not be dropped from queues to keep them short. The streams of these applications have a known, bounded bit rate. Check Point QoS can be configured to forward as much traffic as the stream delivers, ensuring only a small number of packets accumulate in the queues.

Low Latency Queuing allows an Administrator to define special classes of service for delay-sensitive applications. Rules under these classes are used with other rules in the QoS Rule Base.

For each Low Latency class defined on an interface, specify a constant bit rate and maximal delay for active directions. Check Point QoS checks packets matched to Low Latency class rules, to prevent them from being delayed longer than their maximal delay. If the maximal delay of a packet is exceeded, the packet is dropped. Otherwise, packets are transmitted at the constant bit rate defined for the Low Latency class to which it belongs. If the constant bit rate of the class is defined correctly (meaning that it is not smaller than the expected arrival rate of the matched traffic), packets are not dropped. When the arrival rate is higher than the specified constant bit rate, packets exceeding this constant rate are dropped, to ensure that those transmitted are within the maximal delay limitations.

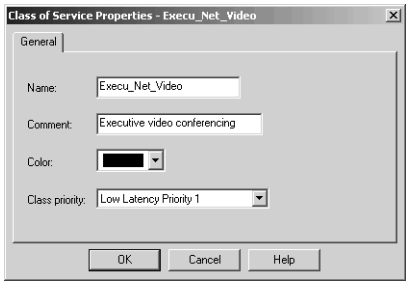
Low Latency Classes

Low Latency classes specify the maximal delay that is tolerated and a constant bit rate. Check Point QoS guarantees traffic matching rules of this type is forwarded within the limits of the bounded delay.

DEFINING A LOW LATENCY CLASS

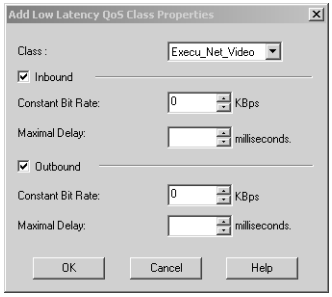
To define a Low Latency Class:

1. Select Manage > QoS > QoS Classes from SmartDashboard menu.
2. Click the New button, and select Low Latency Class of Service.
3. Give the class a name, and assign it a priority.



Low Latency Class Properties

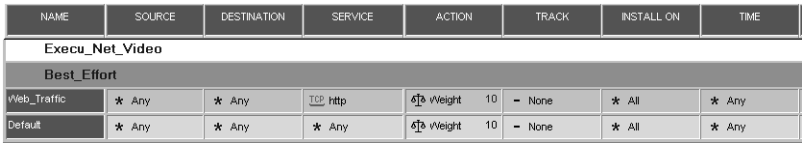
4. Select the target QoS Module. Go to the Topology screen, and select the interface to edit.
5. Click the Add button, and select Low Latency Classes.
6. Populate the Add Low Latency QoS Class Properties screen, and click OK to exit the properties screens.



Low Latency QoS Class Properties

7. Right-click the QoS Rule Base, and select Add Class of Service > Above.

8. Select the target class of service from the Add Class of Service screen, and click OK. The Class of Service Rule is added to the QoS Rule Base. A Best Effort Rule is also added:



NAME	SOURCE	DESTINATION	SERVICE	ACTION	TRACK	INSTALL ON	TIME
Execu_Net_Video							
Best_Effort							
vWeb_Traffic	* Any	* Any	TCP http	vWeight 10	- None	* All	* Any
Default	* Any	* Any	* Any	vWeight 10	- None	* All	* Any

Low Latency Class Rule and Best Effort Rule

9. To activate the Low Latency Class, define at least one rule under it in the QoS Policy Rule Base:

NAME	SOURCE	DESTINATION	SERVICE	ACTION	TRACK	INSTALL ON	TIME
Execu_Net_Video							
Video_Rule	* Any	* Any	* Any	vWeight 10	- None	* All	* Any
Best_Effort							
vWeb_Traffic	* Any	* Any	TCP http	vWeight 10	- None	* All	* Any
Default	* Any	* Any	* Any	vWeight 10	- None	* All	* Any

Low Latency Class Rule with QoS Policy Rule

The traffic matching any Low Latency Class rule receives the delay and constant bit rate properties defined for the specified class, and is also allocated bandwidth according to the rule properties (weight, limit, and guarantee).

The maximal delay is an upper limit. Packets matching the class are always forwarded with a delay not greater, but often smaller, than specified.

Low Latency Class Priorities

It is advisable to define more than one Low Latency Class, if different types of traffic require different maximal delays. Low Latency Classes are assigned one of five priority levels. These priority levels are relative to other Low Latency Classes. The class with the lower maximal delay should get a higher priority than the class with the higher delay. When two packets are ready to be forwarded, one for each Low Latency Class, the packet from the higher priority class is forwarded first. The remaining packet, from the lower class, encounters greater delay.

The maximal delay that can be set for a Low Latency Class depends on the Low Latency Classes of higher priority. Other Low Latency Classes can affect the delay incurred by a class, and must be taken into consideration when setting the minimal delay for the class. Set the priorities for all Low Latency Classes according to maximal delay, then define the classes by descending priority.

For each direction of an interface (inbound and outbound), the sum of the constant bit rates of all the Low Latency Classes cannot exceed 20 percent of the total designated bandwidth rate. The 20-percent limit is set to ensure that Best Effort traffic does not suffer substantial delay and jitter, as a result of the Low Latency Classes.

COMPUTING CONSTANT BIT RATE

To accurately set the constant bit rate of a Low Latency Class, an Administrator must know:

- The bit rate of a single application stream of traffic matching the class.
- The expected number of simultaneously open streams.

The constant bit rate of the class should be the bit rate of a single application, multiplied by the expected number of simultaneous streams:

*(Single Stream) * (Number of Streams Expected) = constant bit rate*

If the number of streams exceeds the number expected when setting the constant bit rate, the total incoming bit rate will exceed the constant bit rate, and many drops will occur. Avoid this situation, by limiting the number of concurrent streams.

COMPUTING MAXIMAL DELAY

The delay value defined for a class determines the number of packets that can be queued in the Low Latency queue, before drops begin. When computing the maximal delay of a Low Latency class, consider both the maximal delay that streams matching the class can tolerate, and the minimal delay that Check Point QoS can guarantee.

Use the following method to estimate the greatest delay the class can tolerate:

1. Refer to the technical details of the streaming application, and find the delay it can tolerate.
2. Find or estimate the delay the external network imposes.
3. Subtract the delay imposed by the external network from the maximum delay the streaming application can tolerate.

Use the following method to estimate the smallest delay for the class:

1. Find the bit rate of the streaming application.

If you cannot find the bit rate of the streaming application in the application properties, you can use SmartView Monitor to observe the application and determine the bit rate.

2. Estimate the typical packet size in the stream.

If you do not know the packet size, you can use `fw monitor` to capture traffic and record packet size, or you can use the LAN's MTU (1500 bytes for Ethernet).

3. Estimate burst size, by monitoring the internal interfaces that precede the QoS Module.

   If no burstiness is detected, the minimum delay of the class should be no smaller than this:

   *(3 * packet size) / bit rate*

If burstiness is detected, the minimum delay of the class should be no smaller than this:

*[(burst size + 1) * packet size] / bit rate*

4. The maximal delay for a class should be between the greatest and smallest delay. Set the delay close to the greatest value, only if the application occasionally bursts.

When you set the maximal delay, you will see an error message, if the value is lower than the value set for other Low Latency Classes of higher priority. Check Point QoS will also display an error message, if the maximal delay is incompatible with interface speed. Both error messages include a minimum acceptable maximal-delay value. Set the maximal delay to a value no smaller than the one printed in the message.

PREVENTING UNWANTED DROPS

If the aggregate bit rate going through the Low Latency Class exceeds the constant bit rate of the class, drops occur. This situation may occur when the number of streams actually opened exceeds the number you expected, when you set the constant bit rate.

Limit the number of connections allowed to the number of connections used to compute the class's constant bit rate, by modifying the QoS Policy as follows:

1. Define a single rule under the class, with a per-connection guarantee as its Action.
2. In the Per Connection Guarantee field of the QoS Action Properties screen, define the per-connection bit rate you expect.
3. In the Number of guaranteed connections field, define the maximum number of connections allowed in this class.
4. Do not select Accept additional non-guaranteed connections.

When to Use Low Latency Queueing

Use Low Latency Queueing when the bit rate of the stream is known and controlling delay is important. Low Latency Classes do not receive TOS markers. If preferential treatment is required beyond the QoS Module, DiffServ should be used instead.

ADVANCED FEATURES

Check Point QoS includes advanced features to allow authenticated QoS, support Citrix MetaFrame, and integrate with Load Sharing configurations.

Authenticated QoS

Check Point Authenticated QoS provides Quality of Service for end-users in dynamic IP environments, such as remote-access and DHCP environments. Authenticated QoS enables priority users, such as corporate CEOs, to receive priority service when remotely connecting to corporate resources.

Authenticated QoS dynamically prioritizes end users, based on information gathered during network or VPN authentication. The feature leverages Check Point UserAuthority technology, to classify both inbound and outbound user connections. Check Point QoS supports Client Authentication, including encrypted Client Authentication, and VPN-1 SecuRemote/SecureClient authentication. User and Session Authentication are not supported.

Citrix MetaFrame Support

Citrix MetaFrame is a client/server software application that enables a client to run a published application on a Citrix server farm, from the client's desktop. One of the disadvantages of using Citrix ICA is that uncontrolled printing traffic can consume all available bandwidth, leaving mission-critical applications struggling.

Check Point QoS solves the problem by:

- Classifying all ICA applications running over Citrix through layer 7.
- Differentiating between the Citrix traffic based on ICA published applications, ICA printing traffic (Priority Tagging), and NFuse.

LIMITATIONS

The following limitations apply when using Check Point QoS to control Citrix MetaFrame traffic:

- The Citrix TCP services are supported in Traditional mode QoS Policies only.
- Session Sharing must be disabled.
- The number of applications that are detected by the inspection infrastructure is limited to 2,048. Console errors will be sent, if this limit is exceeded. These errors are harmless, and will not affect your system. Simply restart the machine.
- Versions of MetaFrame prior to 1.8 are not supported, because there is no packet tagging in these versions.
- Only one Citrix TCP service can be allocated per single rule.

Load Sharing

Load Sharing is a mechanism that distributes traffic within a cluster of Gateways, so the total throughput of multiple machines is increased. Check Point QoS's architecture guarantees that Load Sharing will provide either:

- Two-way stickiness. All packets of a single connection use the same machine in both directions.
- Conversation stickiness. All packets of control/data connections within a conversation use the same machine in both directions.

Check Point QoS provides a fault-tolerant QoS solution for cluster Load Sharing that deploys a unique, distributed WFQ bandwidth-management technology. The user is able to specify a unified QoS Policy per virtual interface of the cluster. The resulting bandwidth allocation is therefore identical to that obtained by installing the same QoS Policy on a single server.

Under a load state, there are a few connections that are backlogged actively for short periods of time. In such cases, the Load Sharing function in ClusterXL is not spread evenly. But in this case, there is no congestion and therefore no need for QoS.

MONITORING QOS POLICY

The QoS Policy and its impact on traffic can be monitored using SmartView Tracker, Eventia Reporter, and SmartView Monitor. Each SmartConsole tool provides a rich set of data, to evaluate the effectiveness of the QoS Policy.

SmartView Tracker

The following two conditions must be met, for a match on a QoS Policy rule to be logged:

- The QoS logging box must be checked on the Gateway Properties > Additional Logging Configuration screen.
- The connection's matching rule must be marked with either Log or Account, in the rule's track field.

QOS TRACK FIELD SET TO LOG

QoS rules with the track field set to Log can generate the following types of log events:

- Connection Rejection

  QoS rejects a connection when the number of guaranteed connections is exceeded, and/or when the rule's action properties are not set to accept additional connections.

- Running Out of Packet Buffers

  QoS generates an Out of Packet Buffers string when QoS global packet buffers are exhausted, or one of the interface direction's packet buffers is exhausted. This report is generated no more frequently than once every 12 hours.

- LLQ Packet Drop

  When a packet is dropped from an LLQ connection, a report is generated. This report is generated no more frequently than once every 5 minutes.

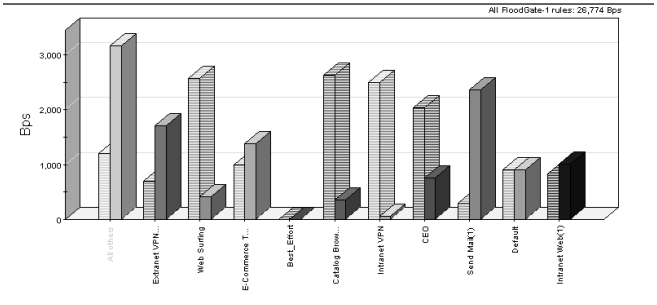
QOS TRACK FIELD SET TO ACCOUNT

QoS rules set to Account generate the following types of events:

- Total bytes transmitted through QoS for each relevant interface and direction
- Total bytes dropped from the connection as a result of the QoS drop Policy
- Count of bytes dropped from the connection, because the maximum used memory fragments for a single connection was exceeded
- Number of bytes dropped from the connection, due to delay expiration for LLQ connections
- Average packet delay for LLQ connections
- Jitter (maximum delay difference between two consecutive packets) for LLQ connections

SmartView Monitor

SmartView Monitor allows an Administrator to view the top QoS rules by interface. Information provided in SmartView Monitor can help an Administrator optimize the QoS Rule Base, by showing which rules should be placed near the top:



Color	Name	Current	Average	Maximum	Minimum	In	Out	Duration
	All others	4,371	9,659	17,633	1,839	1,206	3,165	0:18:36
	Extranet VPN (Day)	2,397	1,716	2,996	0	695	1,702	0:18:24
	Web Surfing	2,995	2,398	2,998	1,801	2,575	420	0:14:40
	E-Commerce Trans.	2,385	1,118	2,997	0	1,001	1,384	0:18:36
	Best_Effort	0	1,956	2,999	0	0	0	0:16:48
	Catalog Browsing	2,990	1,693	2,998	0	2,631	359	0:18:36
	Intranet VPN	2,551	1,559	2,994	0	2,499	52	0:17:36
	CEO	2,791	1,371	2,988	0	2,030	761	0:18:36
	Send Mail(1)	2,659	1,684	2,983	0	292	2,367	0:18:36
	Default	1,806	2,365	2,998	0	903	903	0:18:08
	Intranet Web(1)	1,839	1,776	2,999	0	827	1,012	0:18:36

Sample of SmartView Monitor QoS Information

Eventia Reporter

The Eventia Reporter Network Activity report presents data about traffic accepted by the Gateway. This report can be used to see network activity, to determine effective allocation of bandwidth.

Specific sections include information regarding:

- Overall traffic characteristics, as well as a breakdown by hour and by date.
- The top network users.
- Top services used.
- Top sources and destinations of network traffic

If an Administrator is concerned that an abundance of HTTP and RealAudio traffic is consuming excessive bandwidth and starving IMAP connections, the Top Network Activity - Services section of the Network Activity report determines if proper application of QoS could improve bandwidth allocation:

Top Network Activity Services					
		Traffic Size			
Index	Service	Number of Connections	% of Total Connections	MB	% of Total Bytes
1	http	4,954	50.81%	164.80	25.42%
2	pop-3	2,332	23.92%	74.70	11.53%
3	https	1,428	14.64%	47.28	7.29%
4	ftp	684	7.01%	351.82	54.28%
5	smtp	353	3.62%	9.58	1.48%
	Total (5)	9,751	100.00%	648.18	100.00%
	Average	1,950	20.00%	129.64	20.00%

Top Network Activity - Services Report

OPTIMIZING CHECK POINT QOS

Check Point QoS performance can be improved by following the suggestions below:

- Upgrade to the newest Check Point QoS version available.
- Install Check Point QoS only on the external interfaces of the QoS Module. Unless you are using limits for inbound traffic, installing Check Point QoS only in the outbound direction will provide you the most functionality and improvements.
- Put more frequent rules at the top of your Rule Base. You can use SmartView Monitor to analyze how much a rule is used.
- Turn per-connection limits into per-rule limits.
- Turn per-connection guarantees into per-rule guarantees.